Compression history detection for MP3 audio

Diqun Yan, Rangding Wang, Jinglei Zhou, Chao Jin and Zhifeng Wang
College of Information Science and Engineering, Ningbo University
Ningbo 315211, China
[e-mail: yandiqun@nbu.edu.cn]
*Corresponding author: Yan Diqun

Abstract

Compression history detection plays an important role in digital multimedia forensics. Most existing works, however, mainly focus on digital image and video. Additionally, the existed audio compression detection algorithms aim to detect the trace of double compression. In real forgery scenario, multiple compression is more likely to happen. In this paper, we proposed a detection algorithm to reveal the compression history for MP3 audio. The statistics of the scale factor and Huffman table index which are the parameters of MP3 codec have been extracted as the detecting features. The experimental results have shown that the proposed method can effectively identify whether the testing audio has been previously treated with single/double/triple compression.

Keywords: Compression detection, MP3 audio, scale factor, Huffman table index

1. Introduction

Audio is a prevalent form in multimedia content [1, 2]. When dealing with music, movies, audio is always present supporting a significant amount of information [3]. However, with the rapid development of audio editing tools, it becomes very easy to tamper an audio file without leaving obvious additive clues. How to verify the audio's authenticity and reliability has become an urgent issue that needs to be solved. Generally, audio forgery mainly involves the following steps. First, the target MP3 audio will be loaded into an audio editing software, and then some parts of the audio are modified. Finally, the modified audio is saved as a new MP3 audio file. In fact, the loading operation is always with decompression and recompression will occur during the saving operation. That is, the forgery for an MP3 audio is always with double compression. Hence, by detecting the compression history of the questionable MP3 audio, it is helpful to identify the forgeries.

Most works on compression history detection are focus on digital images [4, 5, 6] and videos [7]. Recently, a few methods for detecting double MP3 compression have been presented. Alessandro et al. [8] achieved the goal of detecting the real bit rate of MP3 audio by analyzing the power spectrum of audio under different bit rates. In [9], the absolute value of proportion for MDCT coefficient which is higher than a fixed threshold was utilized as the detecting feature. Qiao et al. [10] proposed a method based on the small value of MDCT coefficients. Yang et al. [11] utilized the Bendford law for detecting the double compressed MP3 audio. Bianchi et al. [12, 13] estimated the double compressed audios using the characteristics of MDCT coefficient distribution histogram. Besides, the encoding parameters, e.g., Huffman table index and scale factor, were used in the solution for double compression detection by Ma et al [14, 15].

However, all the existing algorithms only focused on detecting whether the suspicious MP3 audio has been compressed once or twice. In a real scenario, the suspicious mp3 might have been double compressed before tampering. In addition, the forgers will decompress the tampered audio and compress it again to eliminate the artifacts left in the audio. As a result, the final tampered MP3 audios are very likely to be compressed more than twice. Therefore, the number of compressions, especially for multiple compressions, is the crucial information to evaluate the audio's reliability.

In this paper, we focused on detecting multiple MP3 compressions. Compared to double compression detection, multiple compression detection is a more challenge problem due to the following reasons. Firstly, for a given audio, with the increase of the number of compressions, the difference between two adjacent compressed audios gets more and more smaller. Secondly, more cases should be considered in detecting multiple compression because there will be more bit rate choices during encoding. The proposed detector is based on the statistical features extracted from MP3 encoder parameters and support vector machine. The experimental results show that the proposed algorithm can effectively detect double and triple MP3 compressions. This rest of this paper is organized as follows. Section 2 briefly reviews the MP3 encoding and introduces the encoding parameters used in this work. The detail of the proposed algorithm is presented in Section 3. In Section 4, the experimental results and discussions are provided and the conclusion is drawn in Section 5.

2. Review of MP3 Encoding

MP3 [16] is still the most prevalent compressed audio format widely used to transfer music over the internet and store audio files on digital audio players. The block diagram of the typical MP3 encoder is shown in **Fig. 1**, which is a two-track structure. On the first track, the audio signal is split into 32 sub bands with the analysis filter bank. The modified DCT is performed in each sub bands, and then 576 MDCT coefficients are obtained for each granule. On the second track, the psychoacoustics model is used in the MP3 standard, which aims to ensure that the added noise is unperceived by the human ear. Here, the audio signal is mapped to the critical band after Fourier transform. Then the masking threshold in each partition is calculated by the frequency and energy of the corresponding audio signal. Since the quantization is operated based on the scale factor band, it needs to convert the masking threshold from frequency band to scale factor band. In this transition, the audio signal is also divided into long or short blocks by the perceptual entropy which is derived from the signal's energy and the masking threshold.
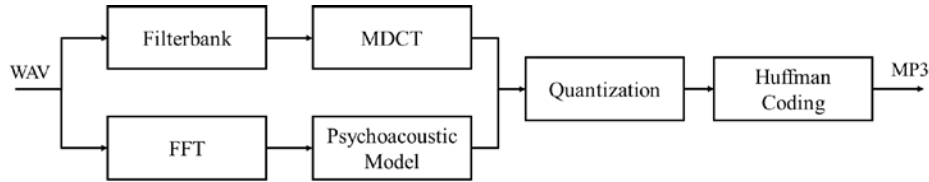

Fig. 1. Block diagram of MP3 encoder

For each long block, the 576 MDCT coefficients are divided into 21 scale factor bands, and the coefficients for each short block are divided into 3 successive bands which consist of 12 scale factor sub bands. After the division, the corresponding minimum allowable distortion is calculated based on the masking threshold and the energy of each scale factor band. Then the MDCT coefficients of each scale factor band are quantized and encoded into bit-stream through the iterative loop which is subdivided into three levels: frame, outer and inner loops. The loops module quantizes an input vector of spectral data in an iterative process according to several demands. The inner loop quantizes the input vector and increases the quantization step size until the output vector can be coded by Huffman coding with the available amount of bit. The quantization of the input vector of spectral values is done according to Eq. (1).

$$ix\left(i\right) = \left\lfloor \left(\left|xr\left(i\right)\right| \middle/ 2^{\frac{p}{4}} \right)^{0.75} \right\rfloor \tag{1}$$

where $xr(i)$ is the i-th MDCT coefficient, $i = 1, 2, \cdots, N$ and $ix(i)$ is the i-th quantized MDCT coefficient (QMDCT). $\lfloor \ \rfloor$ means taking the nearest integer values, and p is the step size for quantization.

The QMDCT coefficients are divided into three regions: big_value region, count1 region and rzero region. Since the coefficients in rzero region are all zeros, the encoder will skip over this region. The other two regions will be coded with 34 Huffman tables. Table 32 and table 33 are for the count1 region. Tables 0 through 31 are used for the big_value region. It should be noted that the table 4 and table 14 are not used in MP3 standard. In addition, the big_value region is further subdivided into three sub-regions (region 0, 1 and 2) and each has its specific

Huffman table. The Huffman table used for each sub-region is determined by the maximum coefficient of the corresponding sub-region.

After the inner loop, an outer loop will check the distortion of each scale factor band. If the allowed distortion is exceeded, the scale factors will be amplified by adding one. the encoder will be back into the inner loop until meeting the bit rate and distortion requirements.

3. Proposed Algorithm

3.1 Motivation

As mentioned in Section 2, lossy compression is used in MP3 standard to remove perceptual redundancy. Most of the high frequency components are the typical redundancy, which is very hard to hear. In [8], it has been found that the high frequency components of MDCT coefficients will be decreasing gradually when the number of compression increases. Thus, compared to single compression, more high frequency components will be removed in double or multiple compressions. It can be inferred that for a given audio frame, the quantization step size will become larger than that in single compression. According to the description in Section 2, the scale factor will decrease and more Huffman table with small indices will be chosen when the step size increases.

Fig. 2 shows the scale factor, Huffman table index and step size extracted from 2000 audios with five genres (Details of the audio are in Section 4). It can be seen that, with the increase of the number of compression, the scale factor decreases and the Huffman table index increases.
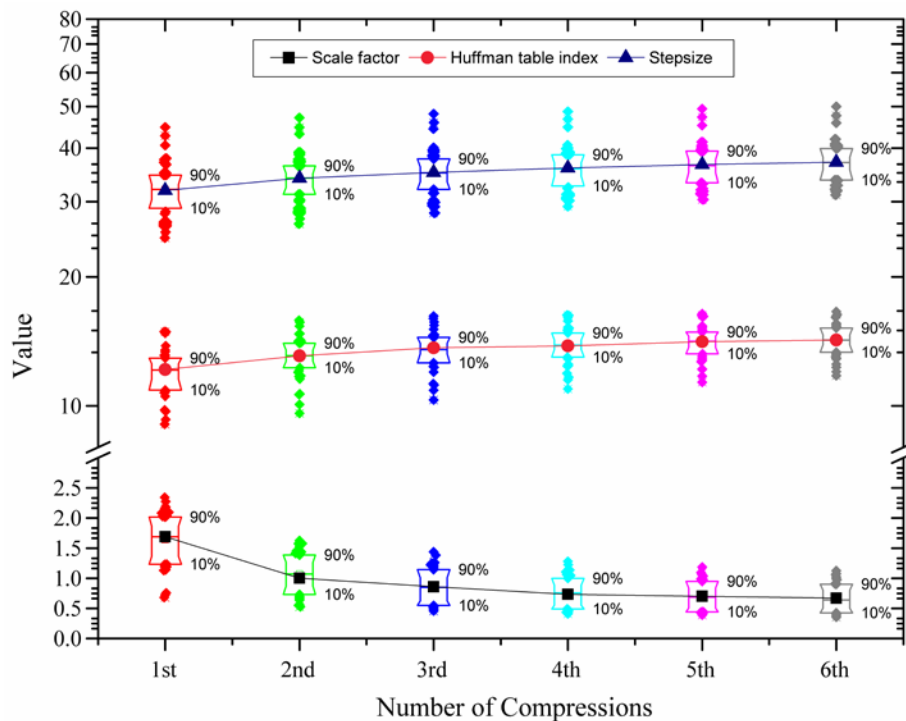

Fig. 2. Encoder parameters with single, double and triple compressions

Moreover, it can be seen from **Fig. 2** that the loss of audio data will become more and more smaller with the increase of the number of compressions. That means that the difference

between double and triple compressed audios is smaller than that between single and double compressed audios. Thus, the statistical differences of these two parameters between the suspicious MP3 audio and its recompressed version could be considered as the evidence to detect the trace of multiple compression.

3.2 Feature Extraction

Three types of features extracted from scale factors and Huffman table indices will be used in the proposed algorithm, differential statistics, probability distribution and cross correlation. As shown in **Fig. 3**, all features are constructed in the following steps. Given an MP3 audio M_1 under examination, we can decompress it into the time domain, and then recompress with the same bit rate as M_1 and obtain a new MP3 audio M_2. The number of frames in M_1 and M_2 is denoted by N. Then the scale factors and Huffman table indices of these two MP3 audios can be obtained by decoding the M_1 and M_2. Finally, the detection features $F_1, F_2, \cdots, F_{16}$ can be calculated by the statistics of these codec parameters.
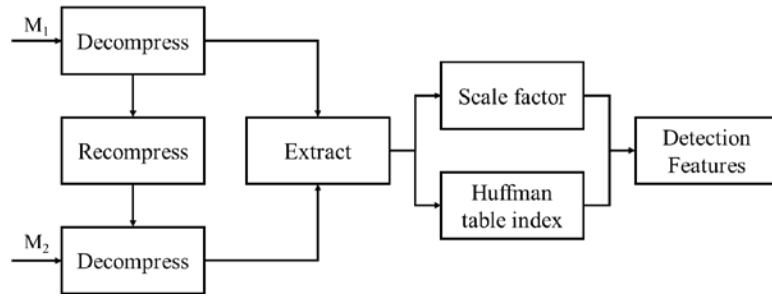

Fig. 3. Block diagram of feature extraction

3.2.1 Differential Statistics

In this work, 21 scale factor bands of long block and 3 sub-regions of big_value region are considered. Thus, for one frame, there will be 21 scale factors and 3 Huffman table indices. Let $s_1(i,j)$ and $h_1(i,k)$ denote the j-th scale factor and the k-th Huffman table index of the i-th frame in M_1 MP3 audio, separately. Here, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, 21$ and $k = 1, 2, 3$. Similarly, $s_2(i,j)$ and $h_2(i,k)$ can be obtained from M_2. Then, the first two features are calculated from M_1 and M_2 over all frames,

$$F_1 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{21}\left[s_1(i,j) - s_2(i,j)\right]}{21 \times N} \tag{2}$$

$$F_2 = \frac{\sum_{i=1}^{N}\sum_{k=1}^{3}\left[h_1(i,k) - h_2(i,k)\right]}{3 \times N} \tag{3}$$

Fig. 4 shows the F_1 and F_2 features obtained from 100 single/double/triple compressed MP3 audios randomly selected from 1000 audios. It can be seen that the two features can effectively distinguish various compression cases.
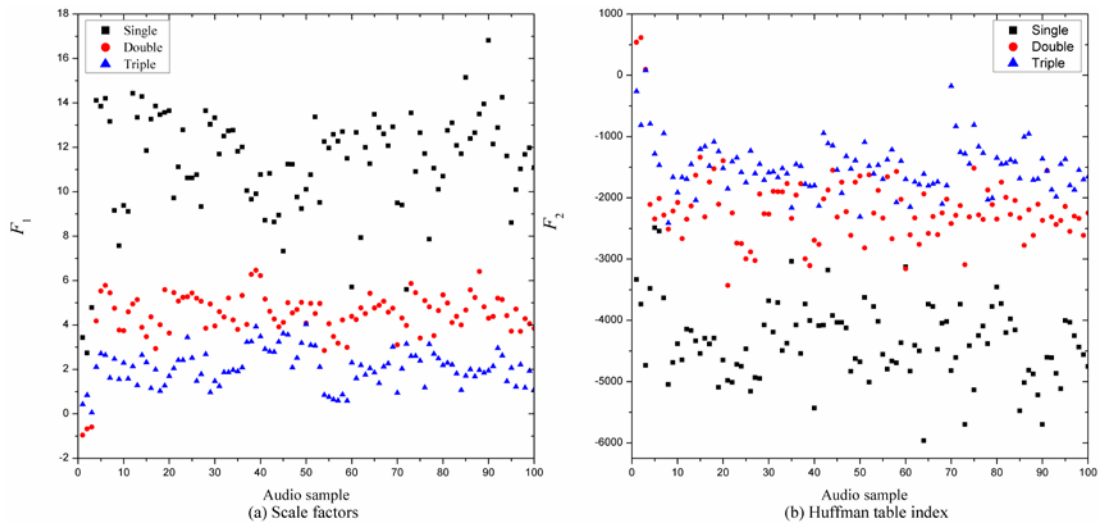

Fig. 4. Differential statistics with the bit rate 128 kbps

3.2.2 Probability Distribution

The histograms of the scale factor and Huffman table indices are shown in **Fig. 5** (a) and (b). Only the scale factors from 0 to 8 are considered in **Fig. 5** (a). It can be seen that, with the increase of the number of compressions, there will be more zero scale factors and Huffman tables with indices 24, 25 and 26.
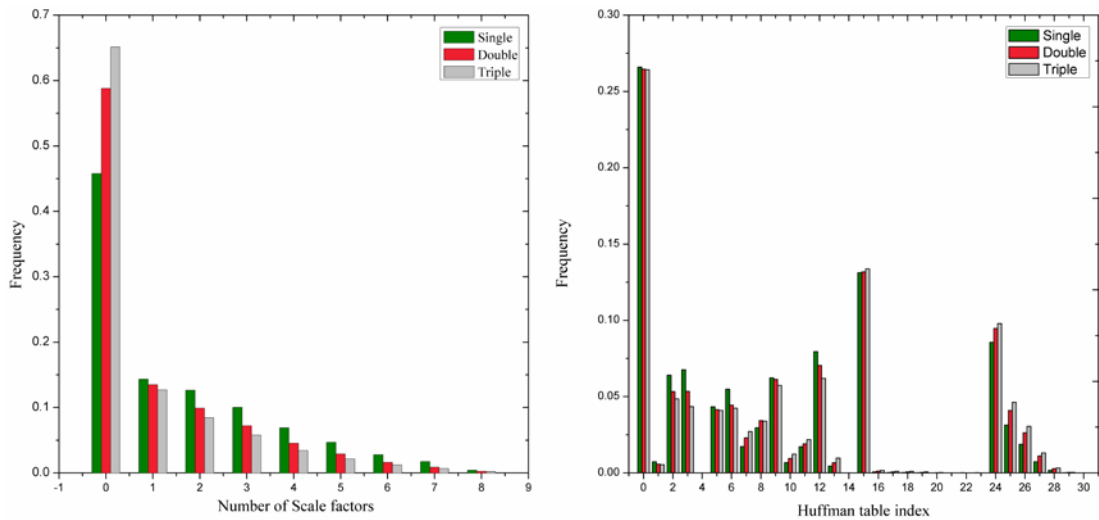

Fig. 5. Histogram of scale factor and Huffman table index (bit rate =128kbps)

Firstly, for M_1 audio, the probability of the specific scale factor and Huffman table index can be calculated by,

$$p_{s,1}(m) = \frac{\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{21}\delta\left(s_1(i,j)==m\right)}{\sum\limits_{k=0}^{8}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{21}\delta\left(s_1(i,j)==k\right)} \tag{3}$$

$$p_{h,1}(r) = \frac{\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{3}\delta\left(h_1(i,k)==r\right)}{\sum\limits_{n=0}^{31}\sum\limits_{i=1}^{N}\sum\limits_{k=1}^{3}\delta\left(h_1(i,k)==n\right)} \tag{4}$$

where $m \in \{0, 1, 2, \cdots, 8\}$ and $r \in \{0, 1, 2, \cdots, 31\}$. $\delta(\cdot) = 1$ if and only if its argument is satisfied, otherwise $\delta(\cdot) = 0$. Similarly, we can obtain $p_{s,2}$ and $p_{h,2}$ from M_2 audio.

Then the third and fourth features can be calculated by,

$$F_3 = p_{s,1} - p_{s,2} \tag{5}$$

$$F_4 = p_{h,1} - p_{h,2} \tag{6}$$
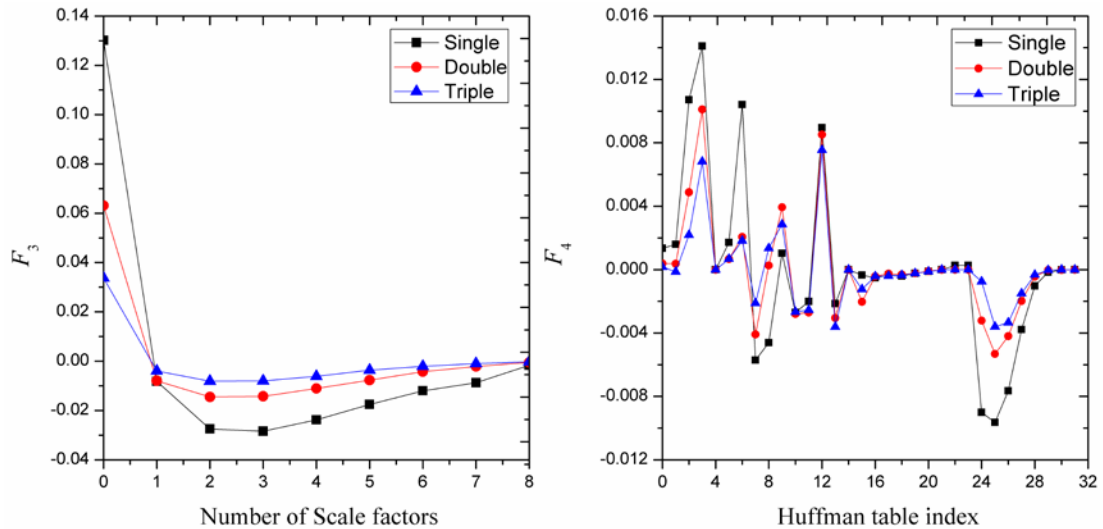
Fig. 6. Increments of probabilities for multiple compressed MP3 audios (bit rate =128kbps)

It can be seen from **Fig. 6** that these two features have good discrimination for various compression cases. As mentioned in Section 2, the 4th and 14th Huffman tables are not used in MP3 encoding. Meanwhile, the values of F_4 for 21th, 22th, 23th, 29th, 30th and 31th are all close to 0. Thus, these 8 elements in F_4 are removed. Finally, the dimensionality of the feature vector is 33 (9+32-8 = 33).

3.2.3 Cross-correlation coefficients

Cross-correlation measures the similarity of two signals and autocorrelation is focused on the correlation of a signal with itself at the different time, which have been widely used as features in pattern recognition applications. In this work, we found that the difference between the cross-correlation and autocorrelation will become more and more smaller when the number of compression increases. Therefore, the statistics such as the mean, variance and Euclidean distance of the difference are extracted as the detecting features.

First, the definitions of normalized autocorrelation α and cross-correlation γ for $x(n)$ and $y(n)$, can be given by,

$$\alpha(w) = \frac{\sum_{n=0}^{N-1} \left(\left(x(n-w) - \overline{x} \right) \left(x(n) - \overline{x} \right) \right)}{\sum_{i=1}^{N} \left(x(n) - \overline{x} \right)^2} \tag{7}$$

$$\gamma(w) = \frac{\sum_{i=1}^{N} \left(\left(x(n) - \overline{x} \right) \left(y(n-w) - \overline{y} \right) \right)}{\sqrt{\sum_{i=1}^{N} \left(x(n) - \overline{x} \right)^2 \sum_{i=1}^{N} \left(y(n-w) - \overline{y} \right)^2}} \tag{8}$$

where $x(n)$ and $y(n)$ is supposed to have N samples, and $\overline{x}$, $\overline{y}$ are the mean or average.

Similarly, the auto-correlation and cross-correlation of the j-th scale factors and k-th Huffman table indices over all M_1 frames can be expressed as $\alpha_{s,1}(j,w)$, $\gamma_{s,1}(j,w)$, $\alpha_{h,1}(k,w)$ and $\gamma_{h,1}(k,w)$. For M_2, $\alpha_{s,2}(j,w)$, $\gamma_{s,2}(j,w)$, $\alpha_{h,2}(k,w)$ and $\gamma_{h,2}(k,w)$ can be calculated. Here, $j = 1, 2, \cdots, 21$ and $k = 1, 2, 3$.

The final auto-correlation features for scale factor are defined by the mean, variance and Euclidean distance,

$$F_3(j) = \frac{1}{N} \sum_{w=1}^{N} \left(\alpha_{s,1}(j,w) - \alpha_{s,2}(j,w) \right) \tag{9}$$

$$F_4(j) = \frac{1}{N} \sum_{w=1}^{N} \left(\alpha_{s,1}(j,w) - \alpha_{s,2}(j,w) - F_3(j) \right)^2 \tag{10}$$

$$F_5(j) = \sqrt{\sum_{w=1}^{N} \left(\alpha_{s,1}(j,w) - \alpha_{s,2}(j,w) \right)^2} \tag{11}$$

Similarly, the cross-correlation features for scale factor F_6, F_7, F_8 can be expressed by $\gamma_{s,1}(j,w)$ and $\gamma_{s,2}(j,w)$. For Huffman table index, we have $F_9 - F_{16}$.

Fig. 7 shows the boxplots of the cross-correlation features for single, double and triple compressed MP3 audios, respectively. The symbol on the horizontal axis in **Fig. 7** denotes the data group extracted at same conditions. For example, *S13* in **Fig. 7**(a) means that this data group is extracted from the 13th scale factors of single-compressed audios. It can be seen that there is an obvious difference on cross-correlation with different compression cases.
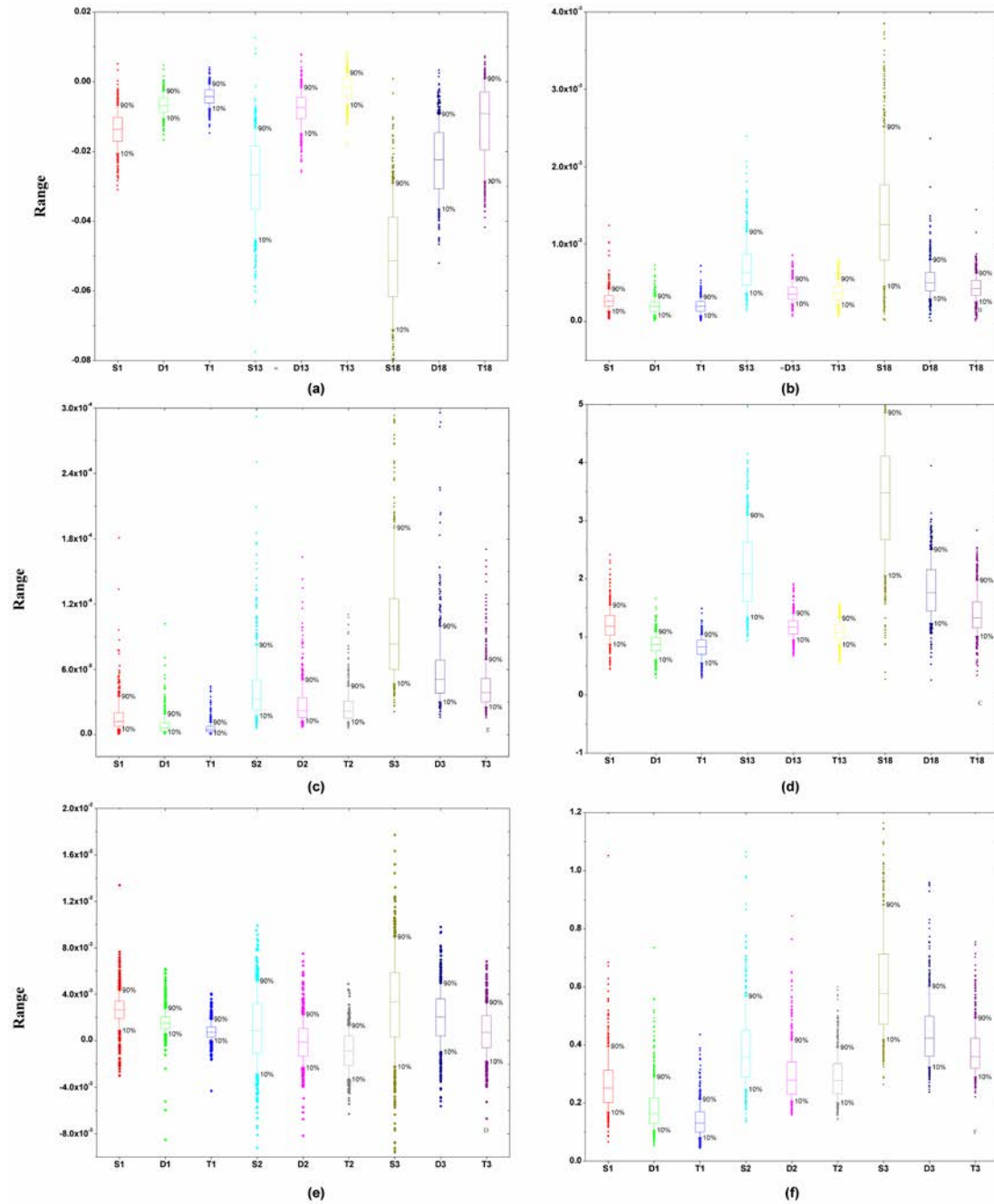
Fig. 7. Cross-correlation features for multiple compressed MP3 audios (bit rate =128kbps).
(a) scale factor, mean; (b) scale factor, variance; (c) scale factor, Euclidean distance; (d) Huffman table index, mean; (e) Huffman table index, variance; (f) Huffman table index, Euclidean distance.

3.3 Classification

The proposed detection algorithm is based on the features defined above and SVM classifier. It consists of a training stage and testing stage. In the training stage, the database for training is composed of a single-compressed audio set and multiple-compressed (double/triple) audio set.

Statistical features above-mentioned are extracted from each audio of the two sets and fed into the SVM to train a classification model. In the testing stage, the features extracted from a testing audio is used as the input feature into the trained model.

4. Experimental Results

An audio set has been built to evaluate the performance of the proposed algorithm. 2000 uncompressed WAV audios are selected randomly as the uncompressed samples, which belong to five different genres: blues, classical, pop, country and folk. The sampling rate of each audio is 44.1 kHz and two different durations, 10 and 20 seconds have been adopted. The information of uncompressed wave audio is given in Table 1. The Lame-3.99.5 MP3 codec (available at http://lame.sourceforge.net) is employed throughout all the experiments. Seven bit-ratios (64, 80, 96, 112, 128, 160 and 192 kbps) are chosen during MP3 compression.

Table 1. Information of uncompressed wave audio

Durations	blues	classical	pop	country	Folk
10	200	200	200	200	200
20	200	200	200	200	200

LIBSVM [17] is used as the SVM classifier and RBF kernel and default parameters have been adopted in training stage. Before training, all features are normalized from -1 to 1. Meanwhile, cross-validation is performed in training model and testing, in which 70% audios in the set are randomly chosen for training a model, and the remaining 30% is used for testing. Every experiment repeats 10 times and the average is taken as the final result.

To evaluate the performance of the proposed algorithm, two experiments have been considered in this work. The first experiment is double compression detection, in which the performances of the proposed method and some existed method [11, 13, 15] are compared. The other experiment is to verify the algorithm's ability to detect triple compression.

4.1 Double Compression Detection

In this case, the single and double compressed samples are chosen as the positive and negative samples, respectively. The experimental results are shown in **Table 2**, in which BR1 and BR2 stand for the bit rates adopted during single and double compression. It can be found that the proposed method is effective for separating single-compressed audio and double-compressed audio when BR2 is larger than BR1.

Table 2. Accuracy of double compression detection by the proposed method (%)

BR1	BR2						
	64	80	96	112	128	160	192
64	95.78	98.07	98.84	99.11	99.93	100	100
80	81.23	94.77	99.67	99.83	98.76	100	100
96	74.90	90.56	98.72	99.13	99.89	99.93	99.89
112	64.53	75.01	76.91	99.14	100	100	100
128	60.77	63.12	64.56	92.91	99.74	99.89	100
160	50.67	52.14	57.09	94.54	97.89	99.45	100
192	50.78	51.23	57.99	93.56	96.32	97.09	98.26

In fact, the bitrate of the testing MP3 audio can be known by parsing the header information. However, the bitrate of the previous compression is blind to us. Therefore, we conduct the following experiment to validate the proposed method's practicability. In this experiment, the single-compressed audio is generated by compressing the non-compressed WAV audio with BR1. For double-compressed audio, firstly, the non-compressed WAV audio is compressed with a bit rate randomly chosen from seven bit rates. And then we decompress it and recompress with BR1 to obtain the double-compressed audio. The experimental results are shown in **Table 3**. In most cases, the proposed algorithm achieves better performance than the other methods. For example, the accuracy of our algorithm is 97.21% at 112kbps bit rate, which is better than other methods.

Table 3. Accuracy of double compression detection (%)

Methods	Bitrate(kbps)						
	64	80	96	112	128	160	192
Proposed	55.35	68.89	75.61	98.09	98.78	99.87	99.91
Yang[8]	53.90	71.67	69.23	95.01	95.67	98.15	99.78
Bianchi[10]	50.77	60.01	61.81	74.08	77.67	86.17	94.44
Ma[12]	53.19	61.78	67.34	91.16	94.23	95.14	96.59

4. 2 Triple Compression Detection

There are several cases during triple compression detection. Let BR1 denote the bit rate used in single-compression, and BR2 for double-compression, BR3 for triple-compression. The results for the case that BR2 is equal to BR3 are shown in **Table 4**. It indicates that the proposed method can effectively detect the compression numbers. Especially when $BR1 \leq BR2 = BR3$, the highest detection rate can reach 100%. It can be observed that the performance of the proposed method is better than the method [12] in the multiple compression detection.

Table 4. Accuracy of multiple compressed detection (Proposed / [12]) (%)

BR1	BR2=BR3						
	64	80	96	112	128	160	192
64	91.09/67.81	95.77/68.51	96.71/72.02	97.28/73.28	98.55/75.09	99.65/77.34	100/75.82
80	90.27/56.55	93.66/61.92	94.06/67.13	97.54/68.95	98.92/73.19	99.07/75.06	100/73.88
96	83.29/56.54	94.74/58.92	95.28/69.03	96.51/71.85	97.09/73.63	98.29/76.41	99.71/74.08
112	75.13/51.17	82.78/55.33	90.27/67.25	94.72/75.12	95.09/79.02	98.42/72.34	99.01/70.01
128	65.17/50.46	73.25/52.88	76.92/56.12	95.15/66.43	97.15/82.71	99.17/70.01	98.82/68.12
160	61.54/50.99	64.52/52.67	68.90/53.74	95.85/59.03	96.33/62.35	98.20/77.41	97.78/73.76

Since there are lots of cases when the BR2 is not equal to BR3, only 64kbps, 96kbps and 128kbps, are considered, and the bit rate of the final compression is set as 128kbps. Hence, for each non-compressed WAV audio, there will be one single-compressed MP3 audio, three double-compressed MP3 audios and nine triple-compressed MP3 audios as shown in **Table 4**.

From **Table 5**, it can be seen that the accuracy of the proposed algorithm is 98.66% when the MP3 audios are single-compressed by 128kbps. It means that most of the 128kbps single-compressed MP3 audios can be effectively discriminated from double and triple compressed MP3 audios. It should be noted that the detection accuracy is very low when the

audios are double compressed by 64kbps and 128kbps. That is that most of the double-compressed audios have been misclassified as the triple-compressed audio. It indicates that the proposed algorithm might lose its effectiveness to detect the compression trace when the audio was compressed from high bitrate to low bitrate, e.g., BR1>BR2, BR1<BR3 and BR2<BR3. Some high frequency components of the audio will be removed when the audio was compressed from high bitrate to low bitrate. This is coincided with the principle of MP3 coding standard. On the contrary, more artificial information will be introduced when the audio is compressed from low bitrate to high bitrate. This is why the detection rate is relatively low at the high-to-low bitrate case.

Table 5. Accuracy of multiple compression detection with the final bit rate is 128kbps (%)

| Final bit rate 128kbps			Compression		
BR1(Kbps)	BR2(Kbps)	BR3(Kbps)	Single	Double	Triple
128	-	-	98.93	1.07	0
64	128	-	0	40.09	61.11
96	128	-	0	37.81	64.38
128	128	-	1.12	91.77	8.23
64	64	128	0	24.12	75.88
64	96	128	0	1.09	98.91
64	128	128	0	2.14	97.86
96	64	128	0	27.44	72.56
96	96	128	0	14.08	85.92
96	128	128	0	3.16	96.84
128	64	128	0	35.97	64.03
128	96	128	0	28.05	71.95
128	128	128	0	4.43	95.57

5. Conclusion

Multiple compressed MP3 audio usually implies a forgery. Therefore, dectection of multiple compression is an important issue in audio forensics. In this paper, an algorithm for MP3 multiple compression has been presented. The detection features are extracted from the statistics of scale factor and Huffman table index. The experimental results show that the proposed algorithm can effectively detect the compression history, especially when the MP3 audio is compressed with $BR1 \leq BR2 \leq BR3$. The robustness of the proposed algorithm will be studied in our future work.

References

[1] R. Hong, M. Wang, M. Xu, T. S. Chua, "Dynamic captioning: video accessibility enhancement for hearing impairment," in *Proc. of ACM International Conference on Multimedia*, pp. 421-430, 2010. Article (CrossRef Link).

[2] R. Hong, M. Wang, G. Li, L. Nie, Z. J. Zha, T. S. Chua, "Multimedia question answering," *IEEE MultiMedia*, vol. 19, no. 4, pp. 72-78, 2012. Article (CrossRef Link).

[3] Y. Chen, T. V. Nguyen, M. Kankanhalli, et al., "Audio maters in visual attention," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 11, pp. 1992-2003, 2014. Article (CrossRef Link).

[4] P. Cecilia, B. Giulia, P. Fernando, "Multiple JPEG compression detection by means of Benford-Fourier coefficients," in *Proc. of IEEE International Workshop on Information Forensics and Security*, pp. 113-118, 2014. Article (CrossRef Link).

[5] S. Milani, P. Bestagini, M. Tagliasacchi, S. Tubaro, "Discriminating multiple jpeg compression using first digit features," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2253-2256, 2012. Article (CrossRef Link).

[6] X. Chu, Y. Chen, M. C. Stamm, K. Liu, "Information theoretical limit of compression forensics," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2689-2693, 2014. Article (CrossRef Link).

[7] S. Milani, P. Bestagini, M. Tagliasacchi, S. Tubaro, "Multiple compression detection for video sequences," in *Proc. of IEEE International Conference on Multimedia Signal Processing*, pp. 112-117, 2012. Article (CrossRef Link).

[8] B. D. Alessandro, Y. Q. Shi, "MP3 bit rate quality detection through frequency spectrum analysis," *ACM Workshop on Multimedia and Security*, pp. 57-62, 2009. Article (CrossRef Link).

[9] Q. Liu, A. Sung, M. Qiao, "Detection of double MP3 compression," *Cognitive Computation*, vol. 2, no. 4, pp. 291-296, 2010. Article (CrossRef Link).

[10] M. Qiao, A. Sung, Q. Liu, "Improved detection of MP3 double compression using content-independent features," in *Proc. of IEEE International Conference on Signal Processing, Communication and Computing*, pp. 1-4, 2013. Article (CrossRef Link).

[11] R. Yang, Y. Q. Shi, J. Huang, "Detecting double compression of audio signal," in *Proc. of SPIE Conference on Media Forensics and Security*, pp. 1-10, 2010. Article (CrossRef Link).

[12] T. Bianchi, A. D. Rosa, M. Fontani, G. Rocciolo, A. Piva, "Detection and classification of double compressed MP3 audio tracks," in *Proc. of ACM Workshop on Information Hiding and Multimedia Security*, pp. 159-164, 2013. Article (CrossRef Link).

[13] T. Bianchi, A. D. Rosa, M. Fontani, G. Rocciolo, A. Piva, "Detection and localization of double compression in MP3 audio tracks," *EURASIP Journal on Information Security*, vol.7, no. 10, pp. 1-14, 2014. Article (CrossRef Link).

[14] P. Ma, R. Wang, D. Yan, C. Jin, "A Huffman table index based approach to detect double MP3 compression," in *Proc. of IEEE International Workshop on Digital-Forensics and Watermarking*, pp. 258-271, 2013. Article (CrossRef Link).

[15] 1 P. Ma, R. Wang, D. Yan, C. Jin, "Detecting Double compression MP3 with the same bit-rate," *Journal of Software*, vol. 9, no. 10, pp. 2522-2527, 2014. Article (CrossRef Link).

[16] ISO/IEC International Standard IS 11172-3, "Information technology - coding of moving pictures and associated audio for digital storage media up to about 1.5 mbit/s - part 3: audio," 1993. Article (CrossRef Link).

[17] C. Chang, C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, 2011. Article (CrossRef Link).
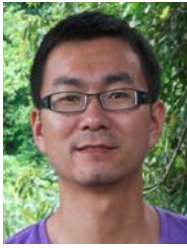
Diqun Yan is an associate professor at Ningbo University, China. He received the Ph. D. from Ningbo University in 2012. His research interests include multimedia forensics and security.

Rangding Wang is a professor at Ningbo University, China. He received the Ph. D. from Tongji University in 2004. His research interests mainly include multimedia security, digital watermarking for digital rights management, data hiding, and steganography.

Jinglei Zhou received her M.S. degree in Communication and Information System. Her research interest is audio forensics.

Chao Jin received his Ph.D. from Ningbo University in 2017. His research interests include audio forensic and steganlaysis.

Zhifeng Wang recevied his B.E. degree from Jiangxi Normal University in 2016. He is currently pursuing a master's degree in Ningbo University. His current research interest is speech forensics.